2D Limit of Dimension-8 Operators Using EFT approach

Israr

December 2022

Preface ...

This is a project report for computational physics class. I would like to say thanks to Prof. Darien Wood for guiding me throughout this project and to Prof. Paul Whitford for giving me this opportunity. I would also like to say thanks to CMS experiment community for allowing me to use all the required resources.

Israr

Abstract

In this project report, I will be discussing about the Dimension-8 operator limits in 2D. Here, Dimension-8 is the mass dimension and 2D means I am finding a sensitive variable with a pair of operators by making a 2D likelihood plot. We have four operators in dim-8 (CBBL4, CBtWL4, CBWL4, CWWL4) which makes 6 pairs. This report contains 5 sections in which I will be discussing about the physics behind the dim-8 operators and ways to find out limits for different variables. I am using 7 variables here defined as met(missing transverse energy), mll(lepton mass), ptl1(transverse momentum of lepton 1), ptl2(transverse momentum of lepton 2), etal1(pseudorapidity of lepton 1), etal2(pseudorapidity of lepton 2). In the end, I am plotting a likelihood plot for a pair of operators and their limits on a bar chart.

Contents

List of Figures

1 Introduction

Effective field theory is a type of approximation which includes the appropriate degree of freedom to describe a physical phenomenon occurring at a certain energy scale. EFT theories allows to systematically parameterize Beyond Standard Model effects and explains how they modify Standard Model processes.The Beyond Standard Model contributions are effectively parameterized in terms of higher order operators with some effective coupling. We define Lagrangian Density in the effective field theory approach as

$$\mathcal{L}_{EFT} = \mathcal{L}_{SM} + \mathcal{L}_{BSM}$$

where $\mathcal{L}_{SM}$ is Lagrangian density of Standard Model and $\mathcal{L}_{BSM}$ is the Lagrangian density for Beyond Standard Model.
I can write $\mathcal{L}_{BSM}$ as :

$$\mathcal{L}_{BSM} = \sum_i \frac{C_i^{(6)}}{\Lambda_i^{(2)}} \mathcal{O}_i^{(6)} + \sum_i \frac{C_i^{(8)}}{\Lambda_i^{(4)}} \mathcal{O}_i^{(8)} + ...,$$

Where C's are coupling constants and $\Lambda' s$ is upper bound on mass/energy. I am interested only in dim-8 operators and their coupling constants.The scattering amplitude would be written in terms of coupling constants and mass limit as :

$$\mathcal{A} = \mathcal{A}_{SM} + \frac{C_\alpha}{\Lambda^4} \mathcal{A}_{Q_\alpha}$$

Taking dim-8 operator as Q Number of events in a given phase space is proportional to the amplitude square of the interaction and can be given as :

$$\mathcal{N} \propto |\mathcal{A}_{SMEFT}|^2 = |\mathcal{A}_{SM}|^2 + \frac{\mathcal{C}_\alpha}{\Lambda^4}.2\mathcal{R}.|\mathcal{A}_{SM}\mathcal{A}^\dagger_{Q_\alpha}| + \frac{\mathcal{C}_\alpha^2}{\Lambda^4}.|\mathcal{A}_{Q_\alpha}|^2$$

Here, I have three terms as standard model term(SM), linear term(Lin)(interaction of standard model and dim-8 operators) and dim-8 operator term(Quad).

To compute these amplitude terms and their coefficients I need to generate few processes and Feynman diagrams corresponding to those processes. I am working with CMS group in ZZ-channel. I am using MadGraph to produce all these processes and their Feynman diagrams. I am using github model (https://github.com/UniMiBAnalyses/D6EFTStudies) which is a model for dim-6 operators and their coupling coefficients as a guide to produce results for dim-8 operators.

2 D8EFTStudies

In this section, I will discuss about the event generation using Madgraph, analysis of data produced from these events,datacards created for these operators to get a likelihood plot, fit and combine and visualization of the results.

2.1 Generation

I am working in ZZ-channel which decays into 4-leptons(electron or muon and their antiparticles with neutrinos associated to them) as shown in figure 1.I am writing a script to generate these processes using Madgraph and all possible Feynman diagrams for these processes.

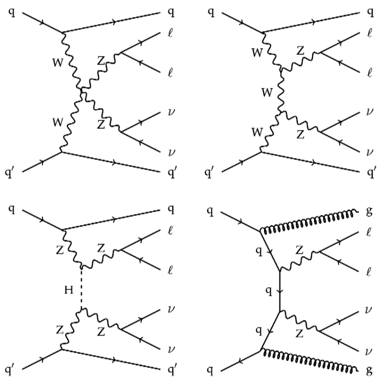
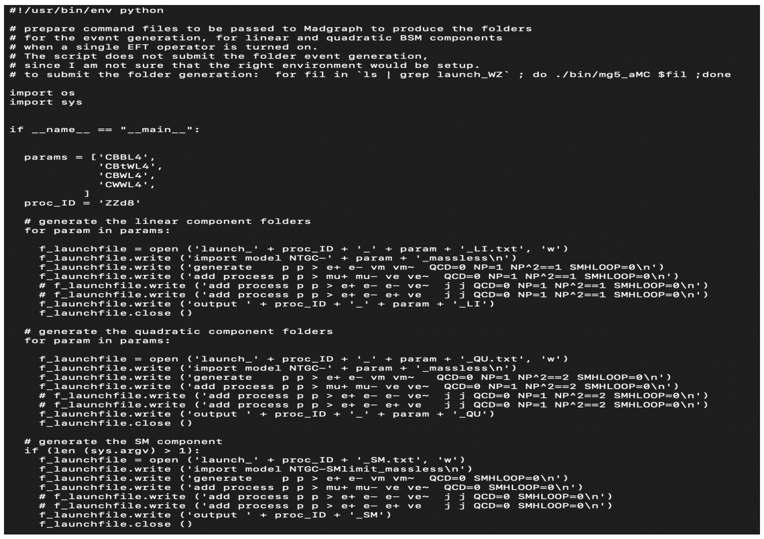


Figure 1: Z-Boson Decay

Here I have four operators in dim-8 (CBBL4, CBtWL4, CBWL4, CWWL4) which are defined under params in the script. I have three different processes as SN,Lin and Quad as discussed earlier. I am using a combination of electron-positron pair with neutrino and antineutrino of muon as one process and muon-antimuon pair with neutrino and antineutrino of electron as other.To do that I have cloned a github model as discussed in introduction.I have created a file with name create1Dfolders_ZZd8.py as shown in figutre 2.This file will generate text files with the name launch_ZZd8.txt. I can run these text files in Madgraph to generate the Feynman diagrams and their amplitude using command -

$$forfilin`ls|greplaunch\_WZ`;do./bin/mg5\_aMC\$fil;done$$

I am using NTGC as UFO model(which includes all the possible interactions among these operators, coupling limits and restriction cards) here to generate all these processes.Similarly,I have written a create2Dfolders_ZZd8.py file to produce all the interaction terms for two different operators as shown in figure 3. After these event generation from Madgraph I have submitted these files to condor and produces LHE files for individual operator(as ZZd8_operator name_LI_results, ZZd8_operator name_QU_results) and their interaction with each other(as ZZd8_operator 1_operator 2_IN_results). All these files are saved into eos-space by making a EFT/LHE link in the generation folder.

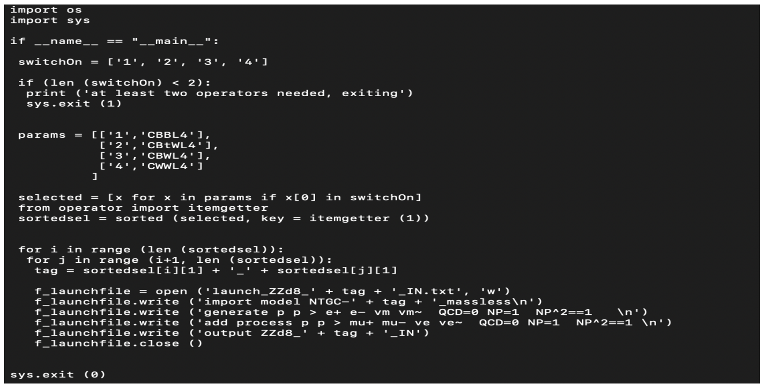


```
#!/usr/bin/env python

# prepare command files to be passed to Madgraph to produce the folders
# for the event generation, for linear and quadratic BSM components
# when a single EFT operator is turned on.
# The script does not submit the folder event generation,
# since I am not sure that the right environment would be setup.
# to submit the folder generation:  for fil in `ls | grep launch_WZ` ; do ./bin/mg5_aMC $fil ;done

import os
import sys


if __name__ == "__main__":


  params = ['CBBL4',
            'CBtWL4',
            'CBWL4',
            'CWWL4',
          ]
  proc_ID = 'ZZd8'

  # generate the linear component folders
  for param in params:

    f_launchfile = open ('launch_' + proc_ID + '_' + param + '_LI.txt', 'w')
    f_launchfile.write ('import model NTGC-' + param + '_massless\n')
    f_launchfile.write ('generate    p p > e+ e- vm vm~  QCD=0 NP=1 NP^2==1 SMHLOOP=0\n')
    f_launchfile.write ('add process p p > mu+ mu- ve ve~  QCD=0 NP=1 NP^2==1 SMHLOOP=0\n')
    # f_launchfile.write ('add process p p > e+ e- e- ve~  j j QCD=0 NP=1 NP^2==1 SMHLOOP=0\n')
    # f_launchfile.write ('add process p p > e+ e- e+ ve   j j QCD=0 NP=1 NP^2==1 SMHLOOP=0\n')
    f_launchfile.write ('output ' + proc_ID + '_' + param + '_LI')
    f_launchfile.close ()

  # generate the quadratic component folders
  for param in params:

    f_launchfile = open ('launch_' + proc_ID + '_' + param + '_QU.txt', 'w')
    f_launchfile.write ('import model NTGC-' + param + '_massless\n')
    f_launchfile.write ('generate    p p > e+ e- vm vm~   QCD=0 NP=1 NP^2==2 SMHLOOP=0\n')
    f_launchfile.write ('add process p p > mu+ mu- ve ve~  QCD=0 NP=1 NP^2==2 SMHLOOP=0\n')
    # f_launchfile.write ('add process p p > e+ e- e- ve~  j j QCD=0 NP=1 NP^2==2 SMHLOOP=0\n')
    # f_launchfile.write ('add process p p > e+ e- e+ ve   j j QCD=0 NP=1 NP^2==2 SMHLOOP=0\n')
    f_launchfile.write ('output ' + proc_ID + '_' + param + '_QU')
    f_launchfile.close ()

  # generate the SM component
  if (len (sys.argv) > 1):
    f_launchfile = open ('launch_' + proc_ID + '_SM.txt', 'w')
    f_launchfile.write ('import model NTGC-SMlimit_massless\n')
    f_launchfile.write ('generate    p p > e+ e- vm vm~  QCD=0 SMHLOOP=0\n')
    f_launchfile.write ('add process p p > mu+ mu- ve ve~  QCD=0 SMHLOOP=0\n')
    # f_launchfile.write ('add process p p > e+ e- e- ve~  j j QCD=0 SMHLOOP=0\n')
    # f_launchfile.write ('add process p p > e+ e- e+ ve   j j QCD=0 SMHLOOP=0\n')
    f_launchfile.write ('output ' + proc_ID + '_SM')
    f_launchfile.close ()
```

Figure 2: Script for Event Generation

2.2 PostProcess and Analysis

I have used a file from the github repository [**1**] with name postprocess.py.The script postProcess.py takes as input a _results folder, controls some basic parameters for the success of the generation, unpacks the LHE files, creates a summary of the run, and cleans the folder from unnecessary log files when called with the clean option (keeping those of failed jobs). It also creates the input cfg file for read_03.cpp, adding the cfg file to the folder itself. read_03.cpp file reads the LHE files and produces ntuples (root files). I have used this file from the analysis folder of [**1**]



```
import os
import sys

if __name__ == "__main__":

 switchOn = ['1', '2', '3', '4']

 if (len (switchOn) < 2):
  print ('at least two operators needed, exiting')
  sys.exit (1)


 params = [['1','CBBL4'],
           ['2','CBtWL4'],
           ['3','CBWL4'],
           ['4','CWWL4']
          ]

 selected = [x for x in params if x[0] in switchOn]
 from operator import itemgetter
 sortedsel = sorted (selected, key = itemgetter (1))


 for i in range (len (sortedsel)):
  for j in range (i+1, len (sortedsel)):
   tag = sortedsel[i][1] + '_' + sortedsel[j][1]

   f_launchfile = open ('launch_ZZd8_' + tag + '_IN.txt', 'w')
   f_launchfile.write ('import model NTGC-' + tag + '_massless\n')
   f_launchfile.write ('generate p p > e+ e- vm vm~  QCD=0 NP=1  NP^2==1    \n')
   f_launchfile.write ('add process p p > mu+ mu- ve ve~  QCD=0 NP=1  NP^2==1 \n')
   f_launchfile.write ('output ZZd8_' + tag + '_IN')
   f_launchfile.close ()


sys.exit (0)
```

Figure 3: Script for Event Generation with interaction of two operators

2.3 Datacards

To build datacards I have cloned a github repository as D6tomkDatacard [**2**] using should have SM ,Lin and Quad term, I have written a s script called ZZd8_to_mkDat.cfg. This file uses ntuples files from EFT/LHE folder in the generation folder to make datacards. information about process name, output folder, operators name, range of operators,variables,cuts on variables and uses ntuples files to make datacards. In figure 4, I have defined 5 section in the script to generate datacards as :

2.3.1 generals

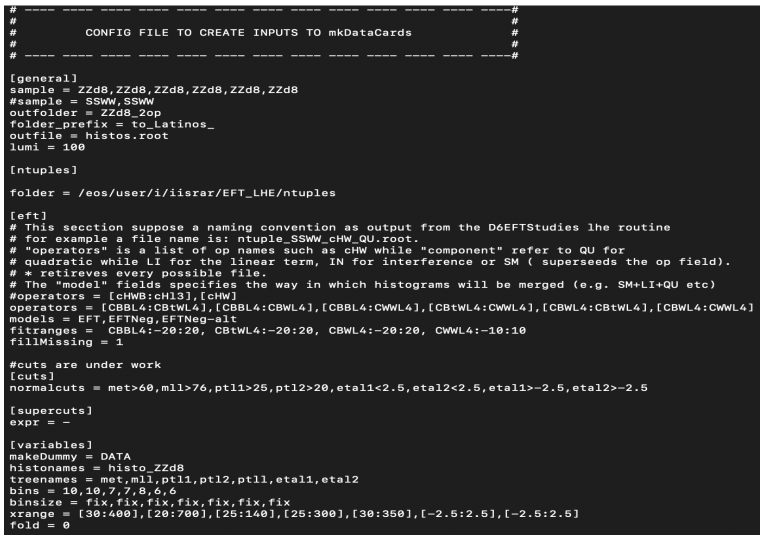
This contains information about the process like if it is dim-6 or dim-8 operators, output root files folder conatining histograms and luminosity.

2.3.2 ntuples

Here, I have defined ntuple files path which contains SM, Lin, Quad and IN result files for each operator and their combination.

2.3.3 eft

Here, I have defined operators and their combination for 2D limits I have 6 combination for 4 operators. I are also adding a range of these operators manually in this section. for simplicity I have used -20:20 for each operator.



```
# ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----#
#                                                                 #
#         CONFIG FILE TO CREATE INPUTS TO mkDataCards             #
#                                                                 #
# ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----#

[general]
sample = ZZd8,ZZd8,ZZd8,ZZd8,ZZd8,ZZd8
#sample = SSWW,SSWW
outfolder = ZZd8_2op
folder_prefix = to_Latinos_
outfile = histos.root
lumi = 100

[ntuples]

folder = /eos/user/i/iisrar/EFT_LHE/ntuples

[eft]
# This secction suppose a naming convention as output from the D6EFTStudies lhe routine
# for example a file name is: ntuple_SSWW_cHW_QU.root.
# "operators" is a list of op names such as cHW while "component" refer to QU for
# quadratic while LI for the linear term, IN for interference or SM ( superseeds the op field).
# * retireves every possible file.
# The "model" fields specifies the way in which histograms will be merged (e.g. SM+LI+QU etc)
#operators = [cHWB:cHl3],[cHW]
operators = [CBBL4:CBtWL4],[CBBL4:CBWL4],[CBBL4:CWWL4],[CBtWL4:CWWL4],[CBWL4:CBtWL4],[CBWL4:CWWL4]
models = EFT,EFTNeg,EFTNeg-alt
fitranges =  CBBL4:-20:20, CBtWL4:-20:20, CBWL4:-20:20, CWWL4:-10:10
fillMissing = 1

#cuts are under work
[cuts]
normalcuts = met>60,mll>76,ptl1>25,ptl2>20,etal1<2.5,etal2<2.5,etal1>-2.5,etal2>-2.5

[supercuts]
expr = -

[variables]
makeDummy = DATA
histonames = histo_ZZd8
treenames = met,mll,ptl1,ptl2,ptll,etal1,etal2
bins = 10,10,7,7,8,6,6
binsize = fix,fix,fix,fix,fix,fix,fix
xrange = [30:400],[20:700],[25:140],[25:300],[30:350],[-2.5:2.5],[-2.5:2.5]
fold = 0
```

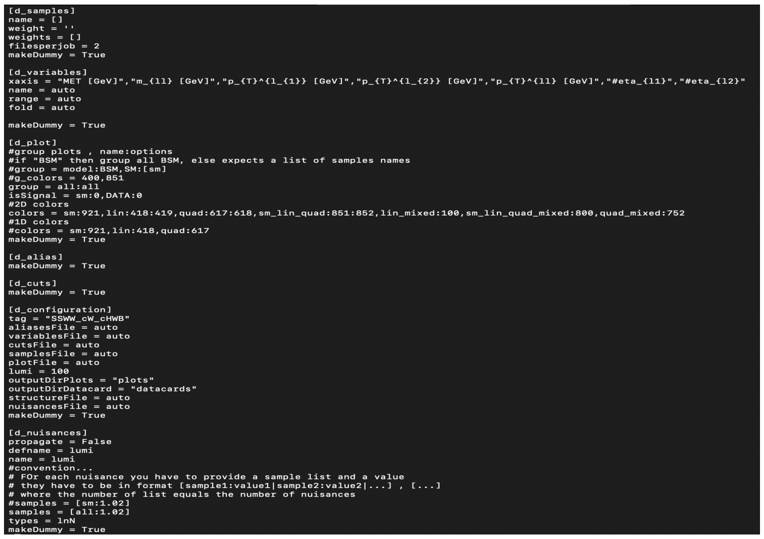
Figure 4: Script for Datacards Generation

2.3.4 cuts and variables

Here, I have defined lower limit on the variables. I am using 7 variables here as met(missing transverse energy), mll(lepton mass), ptl1(transverse momentum of lepton 1), ptl2(transverse momentum of lepton 2), etal1(pseudorapidity of lepton 1), etal2(pseudorapidity of lepton 2). I am also defining number of bins, bin size and range for these variables to make histogram.
I have also put limits on variables and their unit to make a histogram as shown in figure 5.I am running this file using commands to get datacard folder as ZZd8_2op -

$$pythonmkDCInputs.py - -cfg\ cfg/ZZd8\_to\_mkDat\_2D.cfg$$

$$./runmkDatacards.py\ \ - -fZZd8\_2op/to\_Latinos\ \#|foutfolder/prefix$$



```
[d_samples]
name = []
weight = ''
weights = []
filesperjob = 2
makeDummy = True

[d_variables]
xaxis = "MET [GeV]","m_{ll} [GeV]","p_{T}^{l_{1}} [GeV]","p_{T}^{l_{2}} [GeV]","p_{T}^{ll} [GeV]","#eta_{l1}","#eta_{l2}"
name = auto
range = auto
fold = auto

makeDummy = True

[d_plot]
#group plots , name:options
#if "BSM" then group all BSM, else expects a list of samples names
#group = model:BSM,SM:[sm]
#g_colors = 400,851
group = all:all
isSignal = sm:0,DATA:0
#2D colors
colors = sm:921,lin:418:419,quad:617:618,sm_lin_quad:851:852,lin_mixed:100,sm_lin_quad_mixed:800,quad_mixed:752
#1D colors
#colors = sm:921,lin:418,quad:617
makeDummy = True

[d_alias]
makeDummy = True

[d_cuts]
makeDummy = True

[d_configuration]
tag = "SSWW_cW_cHWB"
aliasesFile = auto
variablesFile = auto
cutsFile = auto
samplesFile = auto
plotFile = auto
lumi = 100
outputDirPlots = "plots"
outputDirDatacard = "datacards"
structureFile = auto
nuisancesFile = auto
makeDummy = True

[d_nuisances]
propagate = False
defname = lumi
name = lumi
#convention...
# FOr each nuisance you have to provide a sample list and a value
# they have to be in format [sample1:value1|sample2:value2|...] , [...]
# where the number of list equals the number of nuisances
#samples = [sm:1.02]
samples = [all:1.02]
types = lnN
makeDummy = True
```

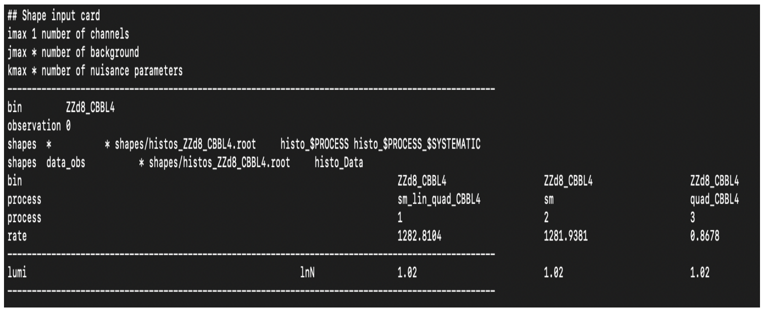
Figure 5: Script for Histogram Generation

3 Fit and Combine

In this section, I will generate all the likelihood plots from the datacards and histograms. To do this I have set-up the cms environment and copied our ZZd8_2op folder to test folder as :

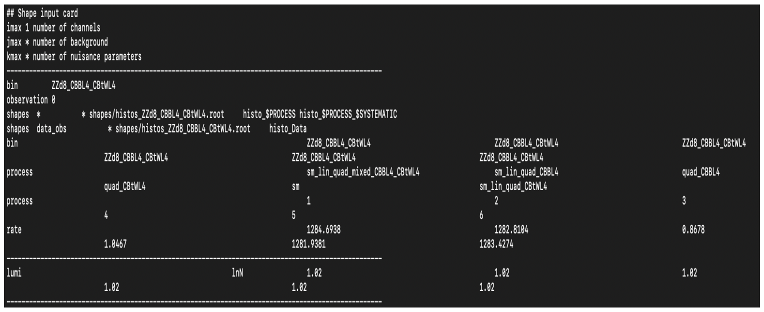
$D6EFTStudies/CMSSW\_10\_2\_13/src/HiggsAnalysis/AnalyticAnomalousCoupling/test$

There is a directory called AnalyticAnomalousCoupling_tests inside test folder which contains a file called createcondor_2D.py. This file takes all the data cards



```
## Shape input card
imax 1 number of channels
jmax * number of background
kmax * number of nuisance parameters
----------------------------------------------------------------------------------------------------
bin        ZZd8_CBBL4
observation 0
shapes  *          * shapes/histos_ZZd8_CBBL4.root     histo_$PROCESS histo_$PROCESS_$SYSTEMATIC
shapes  data_obs          * shapes/histos_ZZd8_CBBL4.root     histo_Data
bin                                                            ZZd8_CBBL4                ZZd8_CBBL4                ZZd8_CBBL4
process                                                        sm_lin_quad_CBBL4         sm                        quad_CBBL4
process                                                        1                         2                         3
rate                                                           1282.8104                 1281.9381                 0.8678
----------------------------------------------------------------------------------------------------
lumi                                                lnN        1.02                      1.02                      1.02
----------------------------------------------------------------------------------------------------
```

Figure 6: 1D Datacard



```
## Shape input card
imax 1 number of channels
jmax * number of background
kmax * number of nuisance parameters
----------------------------------------------------------------------------------------------------
bin        ZZd8_CBBL4_CBtWL4
observation 0
shapes  *          * shapes/histos_ZZd8_CBBL4_CBtWL4.root     histo_$PROCESS histo_$PROCESS_$SYSTEMATIC
shapes  data_obs          * shapes/histos_ZZd8_CBBL4_CBtWL4.root     histo_Data
bin                                                      ZZd8_CBBL4_CBtWL4                        ZZd8_CBBL4_CBtWL4                        ZZd8_CBBL4_CBtWL4
                  ZZd8_CBBL4_CBtWL4                    ZZd8_CBBL4_CBtWL4                        ZZd8_CBBL4_CBtWL4
process                                                  sm_lin_quad_mixed_CBBL4_CBtWL4           sm_lin_quad_CBBL4                        quad_CBBL4
                  quad_CBtWL4                          sm                                       sm_lin_quad_CBtWL4
process                                                  1                                        2                                        3
                  4                                    5                                        6
rate                                                     1284.6938                                1282.8104                                0.8678
                  1.0467                               1281.9381                                1283.4274
----------------------------------------------------------------------------------------------------
lumi                                        lnN          1.02                                     1.02                                     1.02
                  1.02                                 1.02                                     1.02
----------------------------------------------------------------------------------------------------
```

Figure 7: 2D Datacard

for each operator combination and generates likelihood plots while running with command as : ./createCondor_2D.py ZZd8_2op to_Latinos ZZd8 20000 EFT-Neg ([file-name, operator-folder, prefix-to-data cards, process-name, npoints, combined-model]. Combined model uses only positive probability distribution and Lin term can be negative. To solve that problem I have already modified amplitude in Datacard as :

$$\mathcal{N} = SM + k * Lin + k^2 * Quad$$

$$\mathcal{N} = SM + k * (SM + Lin + Quad) - k * (SM + Quad) + k^2 * Quad$$

$$\mathcal{N} = (1 - k) * SM + k * (SM + Lin + Quad) + (k^2 - k) * Quad$$

Therefore, this equation contains all the positive terms as I have shown in the datacards (figures 6 and 7) and I can submit this to combined software to produce a likelihood plot (as shown in figure 8) for each operator combination.

4 Visualization and Results

After producing likelihood plots I am now finding one sigma and two sigma limit of the operator pairs. To do this,I am running a python file called varSens_2D.py

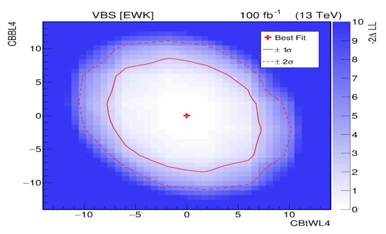
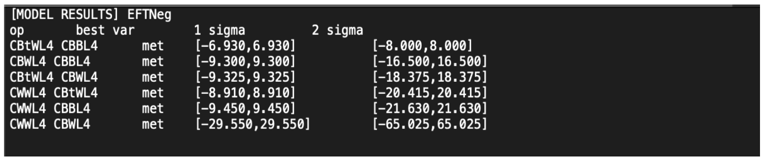


Figure 8: Likelihood Plot for operator combination (CBBL4-CBtwL4)

using the command (python varSens_2D.py –baseFolder SSWW_2op –prefix to_Latinos –lumi 100 –maxNLL 10 –models EFTNeg, EFTNeg-alt –drawText –o sensPlots_SSWW_2op). In this file first I are finding the most sensitive variable for a single operator and then for a pair of operators. After that I are taking the value of that pair of operator and the most sensitive variable for that pair and making a table as shown in figure 9. I have attcahed this varSens_2d,py file in the end of this report.



```
[MODEL RESULTS] EFTNeg
op        best var        1 sigma          2 sigma
CBtWL4 CBBL4      met     [-6.930,6.930]          [-8.000,8.000]
CBWL4 CBBL4       met     [-9.300,9.300]          [-16.500,16.500]
CBtWL4 CBWL4      met     [-9.325,9.325]          [-18.375,18.375]
CWWL4 CBtWL4      met     [-8.910,8.910]          [-20.415,20.415]
CWWL4 CBBL4       met     [-9.450,9.450]          [-21.630,21.630]
CWWL4 CBWL4       met     [-29.550,29.550]        [-65.025,65.025]
```

Figure 9: Limit of a pairs of operators with most sensitive variable

Also,I are plotting these values on a bar chart for a visual representation of sensitivity of operator pairs with a single variable as shown in figure 10.

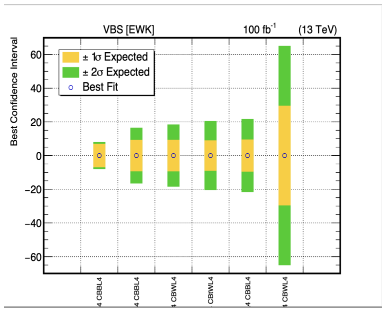


Figure 10: Plot for Limit of a pairs of operators with most sensitive variable

5 Reference

1. https://github.com/UniMiBAnalyses/D6EFTStudies
2. https://github.com/GiacomoBoldrini/D6tomkDatacard
3. https://github.com/latinos/LatinoAnalysis
4. https://github.com/GiacomoBoldrini/AnalyticAnomalousCoupling_tests
5. https://github.com/amassiro/AnalyticAnomalousCoupling
6. Kalinowski, J., Kozów, P., Pokorski, S. et al. Same-sign WW scattering at the LHC: can we discover BSM effects before discovering new states?. Eur. Phys. J. C 78, 403 (2018). https://doi.org/10.1140/epjc/s10052-018-5885-y
7. Chaudhary, G., Kalinowski, J., Kaur, M. et al. EFT triangles in the same-sign WW scattering process at the HL-LHC and HE-LHC. Eur. Phys. J. C 80, 181 (2020). https://doi.org/10.1140/epjc/s10052-020-7728-x